AMENDMENT OF THE CLAIMS

The listing of claims below replaces all prior versions, and listings, of claims:

1. (Cancelled)

2. (Previously Presented) The method of claim 7, wherein receiving a data transaction request comprises receiving a request for loading data into the database.

3. (Previously Presented) The method of claim 7, wherein receiving a data transaction request comprises receiving a request to perform a data transformation operation upon the data in the database.

4. (Original) The method of claim 3, wherein receiving a request to perform the data transformation operation comprises receiving a request to perform one of a data selection operation, a data validation operation, a data cleansing operation, and a data query operation.

5. (Cancelled)

6. (Previously Presented) The method of claim 7, comprising executing a first parallel task in a first number of phases and a second parallel task in a second, different number of phases.

7. (Previously Presented) A method of performing parallel data operations upon data in a database, comprising:

receiving a data transaction request in a client system;

executing a plurality of multi-phase parallel tasks in response to the request to perform the data operations upon the data in the database, wherein executing the multi-phase parallel tasks comprises executing each of the parallel tasks in plural phases; and

each parallel task providing a code to indicate if the task is to be re-invoked in the next phase.

8. (Original) The method of claim 7, wherein providing the code comprises providing the code to a task coordinator.

9. (Original) The method of claim 8, wherein the code comprises a first code to indicate that the task coordinator is to invoke a component in the next phase.

10. (Original) The method of claim 8, wherein the code comprises a second code to indicate that the task is not to invoke a component in the next phase.

11. (Currently Amended) A method of performing parallel data operations upon data in a database, comprising:

receiving a data transaction request in a client system;

executing a plurality of multi-phase parallel tasks in response to the request to perform the data operations upon the data in the database;

analyzing the transaction request;

creating a task plan in response to the transaction request;

implementing the task plan in a multi-phase organization, wherein the plurality of multi-phase parallel tasks are executed to implement the task plan;

~~executing a plurality of tasks in parallel to implement the task plan;~~

determining whether an additional phase is required to execute the tasks based on codes returned by the tasks; and

scheduling an additional phase in response to the determination that an additional phase is required.

12. (Original) The method of claim 11, wherein implementing the task plan comprises creating a job script.

13. (Currently Amended) The method of claim 11, wherein implementing the task plan comprises:

translating the task plan;

selecting a plurality of software components corresponding to the parallel tasks to implement the translated task plan;

assigning a plurality of processes corresponding to the software components; and

creating a communications channel to allow for communications between the processes.

14. (Previously Presented) The method of claim 13, wherein selecting the plurality of software components to implement the translated task plan comprises selecting the plurality of software components to perform at least one of a data extraction operation, a data transformation operation, and a data loading operation.

15. (Previously Presented) An apparatus, comprising:

a user interface;

a processor coupled with the user interface, wherein the processor receives a data transaction request from the user interface; and

a controller coupled with the processor, wherein the controller performs a plurality of tasks in parallel based upon instructions received from the processor, each task performed in a plurality of phases,

each task to provide a code to indicate whether the task is to be re-invoked in a next phase.

16. (Original) The apparatus of claim 15, wherein the processor generates a task plan in response to the data transaction request.

17. (Original) The apparatus of claim 16, wherein the controller comprises a task coordinator to execute the task plan.

18. (Original) The apparatus of claim 16, wherein the controller further comprises a plurality of components to implement the task plan in parallel.

19. (Original) The apparatus of claim 18, further comprising a communications interface for enabling communications between the components.

20. (Original) The apparatus of claim 18, wherein the controller further comprises a storage unit for storing methods and functions to execute the task plan.

21. (Currently Amended) The apparatus of claim 15, ~~wherein the controller is coupled with the processor,~~ wherein the controller performs a number of tasks in parallel based upon instructions received from the processor, each task performed in a plurality of phases further comprises the controller performing the tasks in a sequence of multiple process steps.

22. (Previously Presented) A system, comprising:

a database system;

a network; and

a client system separate from the database system and coupled to the database system over the network, the client system to establish plural sessions with the database system to implement a plurality of data operations upon the database system in parallel.

23. (Cancelled)

24. (Previously Presented) The system of claim 22, wherein the database system is a parallel database system.

25. (Previously Presented) The system of claim 22, wherein the client system comprises:

a processor to receive a data transaction request;

a plurality of operators to perform parallel data operations in response to the data transaction request;

an operator interface coupled to the operators, wherein the operator interface allows communications between the operators.

26.-27. (Cancelled)

28. (Previously Presented) An article comprising at least one storage medium containing instructions that when executed cause a client system to:

receive a data transaction request;

establish plural sessions with a database system over the network connection in response to the request; and

execute a plurality of parallel tasks in the plural sessions to perform data operations upon the data in the database system over a network connection, wherein the client system is separate from the database system.

29. (Previously Presented) The article of claim 28, wherein the instructions when executed cause the client system to execute each of the parallel tasks in plural phases.

30. (Previously Presented) The article of claim 29, wherein the instructions when executed cause the client system to execute a first parallel task in a first number of phases and a second parallel task in a second, different number of phases.

31. (Original) The article of claim 29, wherein the instructions when executed cause each parallel task to provide a code to indicate if the task is to be re-invoked in the next phase.

32. (Original) The article of claim 31, wherein the instructions when executed cause the parallel task to provide the code to a task coordinator.

33. (Original) The article of claim 32, wherein the code comprises a first code to indicate that the task coordinator is to invoke a component in the next phase.

34. (Original) The article of claim 32, wherein the code comprises a second code to indicate that the task is not to invoke the component in the next phase.

35.-36. (Cancelled)

37. (Previously Presented) The method of claim 38, comprising executing a first parallel task in a first number of phases and a second parallel task in a second, different number of phases.

38. (Previously Presented) A method of performing parallel data operations upon data in a database, comprising:

receiving a data transaction request; and

executing a plurality of synchronized multi-phase parallel tasks in response to the request to perform the data operations upon the data in the database,

wherein executing the multi-phase parallel tasks comprises executing each of the parallel tasks in phases; and

each parallel task providing a code to indicate if the task is to be re-invoked in the next phase.

39. (Original) The method of claim 38, wherein providing the code comprises providing the code to a task coordinator.

40. (Original) The method of claim 39, wherein the code comprises a first code to indicate that the task coordinator is to invoke a component in the next phase.

41. (Original) The method of claim 39, wherein the code comprises a second code to indicate that the task is not to invoke a component in the next phase.

42. (Previously Presented) The method of claim 39, wherein the code synchronizes the operation of one or more components.

43. (Previously Presented) The method of claim 7, wherein executing the plurality of multi-phase parallel tasks comprises:

executing at least first and second software components in parallel;

each of the first and second software components performing one or more operations in a first phase;

waiting for a message comprising the code from each of the first and second software components prior to proceeding to a second phase; and

each of the first and second software components performing one or more operations in the second phase.

44. (Previously Presented) The method of claim 43, further comprising:

waiting for another message from each of the first and second software components prior to proceeding to a third phase;

the first software component performing one or more operations in the third phase; and

the second software component being idle in the third phase.

45. (Previously Presented) The method of claim 44, further comprising:

receiving a first message from the first software component indicating that the first software component is to be re-invoked in the third phase; and

receiving a second message from the second software component indicating that the second component is not to be re-invoked in the third phase.

46. (Previously Presented) The apparatus of claim 15, wherein the controller comprises at least first and second software components executable in parallel to perform the plurality of tasks;

wherein each of the first and second software components is executable to perform one or more operations in a first phase;

the controller to wait for a message comprising the code from each of the first and second software components prior to proceeding to a second phase; and wherein each of the first and second software components is executable to perform one or more operations in the second phase.

47. (Previously Presented) The apparatus of claim 46, wherein the controller is adapted to further wait for another message from each of the first and second software components prior to proceeding to a third phase;

wherein the first software component is executable to perform one or more operations in the third phase, and the second software component is idle in the third phase.

48. (Previously Presented) The apparatus of claim 47, wherein the controller is adapted to further:

receive a first message from the first software component indicating that the first software component is to be re-invoked in the third phase; and

receive a second message from the second software component indicating that the second component is not to be re-invoked in the third phase.

49. (Previously Presented) The system of claim 22, wherein the client system is adapted to execute plural tasks in parallel, each of the plural tasks executable in plural phases.

50. (Previously Presented) The article of claim 29, wherein executing each of the parallel tasks in plural phases comprises:

executing at least first and second software components in parallel;

each of the first and second software components performing one or more operations in a first phase;

waiting for a message from each of the first and second software components prior to proceeding to a second phase; and

each of the first and second software components performing one or more operations in the second phase.

51. (Previously Presented) The article of claim 50, wherein the instructions when executed cause the client system to further:

wait for another message from each of the first and second software components prior to proceeding to a third phase;

cause the first software component to perform one or more operations in the third phase; and

cause the second software component to be idle in the third phase.

52. (Previously Presented) The article of claim 51, wherein the instructions when executed cause the client system to further:

receive a first message from the first software component indicating that the first software component is to be re-invoked in the third phase; and

receive a second message from the second software component indicating that the second component is not to be re-invoked in the third phase.

53. (New) The method of claim 7, wherein executing the plurality of multi-phase parallel tasks includes executing a plurality of checkpoint tasks in parallel, each checkpoint task having multiple phases, and each checkpoint task to write data to storage to provide an indication of a current execution point.

54. (New) The method of claim 7, wherein executing the plurality of multi-phase parallel tasks includes executing the plurality of multi-phase tasks in parallel.

55. (New) The apparatus of claim 15, wherein the plurality of tasks include a plurality of checkpoint tasks that are executed in parallel, each checkpoint task having multiple phases, and each checkpoint task to write data to storage to provide an indication of a current execution point.